DRAFT

DEFENSE INFORMATION SYSTEMS AGENCY

JOINT INTEROPERABILITY TEST COMMAND
FORT HUACHUCA, ARIZONA

VALIDATION TEST REPORT

FOR

12-BIT JOINT PHOTOGRAPHIC EXPERTS GROUP POST PROCESSING AND FORWARD ERROR CORRECTION FOR THE NATIONAL IMAGERY TRANSMISSION FORMAT STANDARD

DRAFT

VALIDATION TEST REPORT

FOR

12-BIT JOINT PHOTOGRAPHIC EXPERTS GROUP POST PROCESSING AND FORWARD ERROR CORRECTION FOR THE NATIONAL IMAGERY TRANSMISSION FORMAT STANDARD

25 February 1997

Submitted by: Jack Brandt
Chief
Open Systems Division

Approved by: ______________________________
STEVE E. BRIDGES
Department Head
C2 and Combat Support Systems Department
Joint Interoperability Test Command

Prepared under the direction of:

Mr. Stephen W. Kerr
Joint Interoperability Test Command
Fort Huachuca, Arizona 85613-7020

EXECUTIVE SUMMARY

At the request of the National Imagery and Mapping Agency (NIMA) (formerly the Central Imagery Office (CIO)), the Joint Interoperability Test Command (JITC) conducted validation testing of the proposed format for Tagged Record Extensions and Joint Photographic Expert Group (JPEG) application markers for Output Amplitude Mapping Methods for incorporation into Military Standard 2500, National Imagery Transmission Format (NITF) version 2.0 and Military Standard 188-198A, Joint Photographic Experts Group (JPEG) Image Compression for the NITFS. The tagged record extensions and JPEG markers support the post-processing of pixel data following 12 bit Joint Photographic Experts Group (JPEG) Discrete Cosine Transform (DCT) decompression, error correction of received bit errors using forward error correction (FEC) data, indexing using image frame offsets, and application of NITF APP_8 and APP_9 data segments.

The JITC NITF Certification Test and Evaluation (CTE) Facility conducted the validation test from December 1995 through June 1996.

The validation test confirmed that the proposed RFC's were implementable and complete enough to incorporate into the NITF Standards, test procedures and test tools were able to measure test criteria, and the sample implementation conformed to the proposed standards. The JITC recommends that the proposed text for the tags IOMAPA, NHECCA, NIECCA and NBLOCA be incorporated in to Military Standard 2500 and the NITFS Tag Register. The proposed text for application data segments #8 and #9 should be incorporated in to Military Standard 188-198.

TABLE OF CONTENTS

PARAGRAPH **PAGE**

TABLE OF CONTENTS (Continued)

LIST OF FIGURES and TABLES

TABLE OF CONTENTS (Continued)

LIST OF FIGURES and TABLES (Concluded)

LIST OF APPENDICES

TABLE OF CONTENTS (Concluded)

LIST OF APPENDICES (Concluded)

SECTION I: INTRODUCTION

1-1 BACKGROUND

1-1.1 General. The National Imagery and Mapping Agency (NIMA) (formerly the Central Imagery Office (CIO)) requested the Joint Interoperability Test Command (JITC) conduct validation testing of the proposed format for tagged record extensions and Joint Photographic Expert Group (JPEG) application markers for Output Amplitude Mapping Methods for incorporation into the Military Standard 2500, National Imagery Transmission Format version 2.0, and Military Standard 188-198A, JPEG Image Compression for the NITFS. The NITFS is a suite of standards for formatting digital imagery related products and exchanging them among members of the Intelligence Community (as defined by Executive Order 12333), the Department of Defense, and other departments or agencies of the United States Government, as governed by Memoranda of Agreement (MOA) with those departments or agencies.

1-1.2 Tagged Record Extension Format. The proposed tagged record extension formats and JPEG application markers support the post-processing of pixel data following 12 bit Joint Photographic Experts Group (JPEG) Discrete Cosine Transform (DCT) decompression. The use of the proposed Input/Output Mapping Methods tag, IOMAPA, will decrease the image quality loss produced by the JPEG/DCT compression for a given compression rate. The use of JPEG application markers APP_8 and APP_9 provide image block minimum values and Forward Error Correction (FEC) protection of JPEG header information.

1-1.3 Validation Methodology. The NITFS validation methodology is based on a five step validation process as outlined in JIEO Circular 9008. Upon successful completion of these steps, the proposed additions to the standards are considered to be validated. A natural outcome of this process is the creation of the Means of Testing (MOT) for testing products for conformance to the new features of the standard.

a. Step 1. First the service, functional, and performance requirements are fully identified and appropriate authority ratifies that the requirements are valid. Next, the test objectives and criteria are developed that will be used to ascertain whether the proposed solution satisfies the validated requirements. As the appropriate authority, the Imagery Standards Management Committee (ISMC)/NITFS Technical Board (NTB) ratified the requirements and the validation objectives and criteria.

b. Step 2. As the proposed addition to the standard is written, conformance test objectives, criteria, and test cases are also written. The conformance test objectives for the proposed tagged record extension formats and application markers are identified in Section 2 of the validation test plan.

c. Step 3. A physical realization of the addition to the standard is implemented. A national program office sponsored the development of simulation software and sample NITF files containing data in the proposed tag and markers. The JITC developed test procedures and software test tools needed to conduct conformance testing independent of the developer, but in coordination with the development of the nominated tags and JPEG application markers.

d. Step 4. The conformance test procedures and tools were used to verify that the sample implementation and sample files conform to the proposed standard. Based on conformance test results, the sample implementation and files were modified and retested until they adequately conformed with the proposed standard. In some instances the proposed standard was modified based on lessons learned from the implementation and testing effort.

e. Step 5. Once the sample implementation is verified as conforming to the proposed standard, the implementation is evaluated against the objectives and criteria defined in the first step to measure how well the proposed standard meets the original service, functional, and performance requirements. During this phase, pair-wise comparisons are made of multiple implementations of the standard, providing assurance that all standard compliant implementations are interoperable. This step was not accomplished within the scope of this plan for this test effort. It is a follow-on action to be taken under the direction of the ISMC/NTB.

1-2 PURPOSE. To determine the validity of the proposed tagged record extensions.

1-3 SCOPE

1-3.1 Overview. Figure 1 portrays the general approach JITC used to validate the proposed tagged record extension formats for the NITFS. The subsections are summarized as follows.

a. Documentation Static Review

(1) Analyze Standards. The first major NITFS validation process reviews and analyzes the entire suite of NITF standards and the proposed additions to the standards to identify any internal conflicts, oversights, or ambiguities which are considered faults and must be resolved before the proposed additions can be verified.

(2) Identify Specified Requirements. A complete set of requirements were extracted from the proposed standard and broken down into those which state policy and those involving implementation issues. Both types of requirements are

important for the overall analysis of the standard; those involving implementation issues are of particular use in nominating the test criteria and strategies.

(3) Nominate Test Criteria and Strategies. Test strategies and methods were identified and nominated by which compressed files and interpret capable implementations can be tested for conformance with each implementation requirement.

b. Implementation Dynamic Review. The second major NITFS validation process is to develop candidate test cases and execute the test strategy on sample files and sample implementations of the proposed standard. To be useful in DOD procurements, conformance to mandated requirements particularly those with operational consequences must be testable. There are several steps to the determination of testability as outlined in the following subsections.

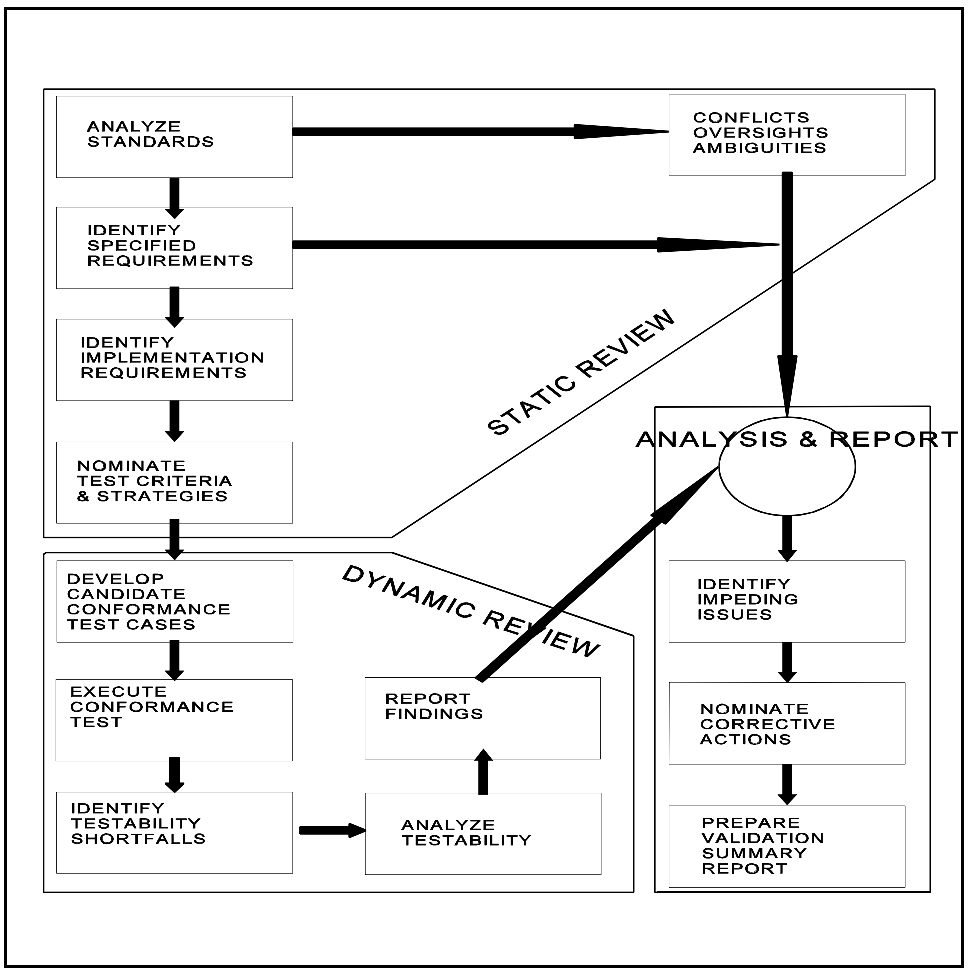

Figure 1 General Validation Approach

(1) Develop Candidate Conformance Test Cases. Based on the identified implementation requirements, candidate conformance test cases were developed for evaluation of requirements.

(2) Execute Conformance Test. Based on the data provided by the developer, the candidate test cases were executed and results were collected.

(3) Identify Testability Shortfalls. All shortfalls on the overall testability of the proposed standard were identified.

(4) Analyze Testability. The conformance test results and testability shortfalls were reviewed for overall impact on testability of the proposed standard. During analysis, all facets of the effort (proposed text, sample implementation, and candidate test cases) were considered suspect.

(5) Report Findings. All problems and shortfalls identified in the dynamic review process were reported for inclusion in the overall analysis of the static and dynamic results of the process.

c. Analysis & Report

(1) Identify Impeding Issues. The static and dynamic reviews resulted in a set of issues which were documented and analyzed for corrective action.

(2) Corrective Actions. Corrective actions for each identified issue were nominated to the preparing agency. Recommendations were made to either change the proposed text of the standard, change the sample implementation, or change the candidate test cases. The objective was to get the proposed standard, sample implementation, and the candidate test cases in harmony. Resolutions to the issues will be again passed through the static and dynamic review cycles.

(3) Validation Report. All of the analyses and associated efforts are documented in this validation report.

1-3.2 Resources. The NITFS Certification Test and Evaluation (CTE) Facility is configured to support testing of digital imagery systems. Figure 2 provides an overview of the CTE Facility's configuration. The developer provided simulation software and sample NITF files containing data in the proposed tagged record extension formats within the image extended subheader data field that were loaded on the CTE Facility's digital imagery systems. They are uniquely qualified to perform this validation assessment of the proposed tagged record extension formats for inclusion in to the NITF format. The NITFS CTE Facility test personnel have previously validated tagged record extension formats for the Defense Mapping Agency's Digital

Point Positioning Data Base (DPPDB) tags and the Profile for Imagery Architecture (PIA) tags. In addition to validating the imagery compression algorithms for JPEG, Bi-Level, Adaptive Recursive Interpolated Differential Pulse Code Modulation (ARIDPCM), and Vector Quantization (VQ), the CTE Facility has conducted numerous tests for imagery military standards over the past four years.

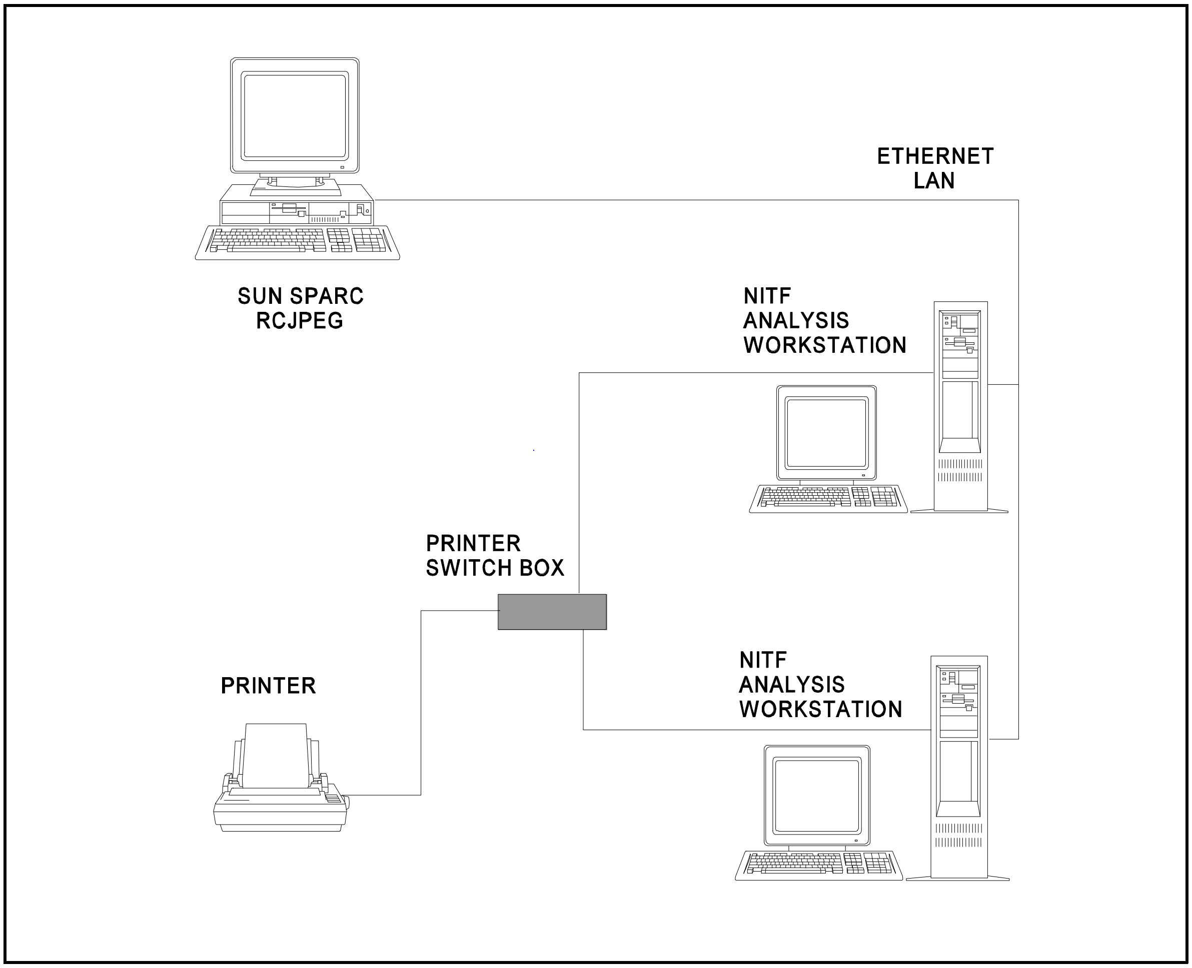

Figure 2 CTE Facility Configuration

1-3.3 Test Schedule. The static portion of the validation began in December 1995 as part of the process needed to prepare the validation plan. The initial phase of conduct for the dynamic review began in January 1996 and was completed in June 1996.

1-3.4 Limitations. Standards written in English cannot be guaranteed to be free of conflicts, unambiguous, and complete. The documentation static review by itself provides a limited level of confidence that the addition to the standard meets these requirements. The implementation review provides an analysis of a sample implementation built in accordance with the text of the proposed standard; determines the extent to which the proposed standard is implementable; determines the extent to which the proposed standard is testable through test case scenario development; and confirms, identifies shortfalls, and/or refutes findings from the documentation static review of the proposed standard. The documentation static review coupled with the implementation dynamic review provides a more thorough analysis of the proposed standard and a higher level of confidence in the validation of the standard than if only the traditional text review were conducted.

1-4 TAG RECORD EXTENSION FORMATS TO BE TESTED. The proposed tags and JPEG segments to be tested included IOMAPA, NHECCA, NIECCA, NBLOCA, APP_8 and APP_9. During the evaluation of the JPEG post processing, the related NITF JPEG Application data segments were renamed. NITF APP_8 segment was changed to APP_6/0001 and NITF APP_9 segment was changed to APP_6/0002. In the final version of the Requests For Change (RFC) to the applicable NITF military standards, the new names are used. To preclude confusion and to maintain synchronization with the validation test plan, this report uses the original segment names.

1-4.1 IOMAPA Tagged Extension Description. The IOMAPA tagged record extension contains the data necessary to generate the output amplitude mapping function which is applied to the image data after it has undergone the data expansion process utilizing the 12 bits-per-pixel JPEG-DCT decompression. Refer to Appendix B for details.

1-4.2 NHECCA Registered Tagged Record Extension Description. The forward error correction (FEC) codes stored in this extension provides acceptable protection to the NITF header from random bit errors in the range of one in 10^6 bits. Refer to Appendix C for details.

1-4.3 NIECCA Registered Tagged Record Extension Description. The FEC codes stored in this extension provide acceptable protection to the image subheader from random bit errors in the range of one in 10^6 bits. Refer to Appendix D for details.

1-4.4 NBLOCA Tagged Record Extension Description. This tagged extension stores the offset of each image frame relative to each other within an NITF image. This extension allows the NITF file to be read in a random access or parallel fashion. Refer to Appendix E for details.

1-4.5 NITF APP_8 Segment Image Block Minimum Values. The NITF APP_8 application data segment contains the image block minimum *uncompressed* value *and appropriate scale factor.* This value is used by the 12 bit JPEG/DCT compression process to store the minimum value of the image block. This minimum value is subtracted from the image block data prior to input to the input amplitude remapping process described in Appendix B, IOMAPA Tagged Record Extension Description, paragraph 30.2.3 Refer to Appendix F for details.

1-4.6 NITF APP_9 Segment Forward Error Correction Code. The NITF APP_9 application data segment contains FEC codes which are used to protect NITF-JPEG/DCT header and table data from bit errors. Refer to Appendix G for details.

SECTION II: DETAILS OF TEST

II-0 GENERAL. The overall test objective is to ensure the proposed tagged record extension formats for Input/Output Amplitude Mapping Methods and Forward Error Correction (FEC) protection can be incorporated into the National Imagery Transmission Format Standards in a manner that is technically correct, consistent, complete, and testable.

2-1 SUBTEST I, DOCUMENTATION STATIC REVIEW

2-1.1 Objective. To determine to what extent the proposed tagged record extension formats and JPEG markers are technically correct, consistent, complete, and can be incorporated within the overall NITF file structure.

2-1.2 Criteria. In the context of standards validation, the term "validation" means to determine whether a standard is capable of supporting its intended use and is adequately documented to support implementation by disparate developers. The intended use of the proposed tagged record extensions is to establish the requirements for the communication or interchange of 12 bpp JPEG/DCT post-processing image data and provide forward error correction for critical data in the file. From this intended use, a number of key criteria can be derived which the standard must be able to support:

a. Free of Conflicts. The proposed addition to MIL-STD-2500 and the complete suite of NITF standards must be mutually free of conflicts (e.g. technical or logical conflicts of requirements, that if implemented, cause the implementation to violate conformance to other related standards or clauses of the same standard) or contains a clear set of precedence statements by which any conflicts can be resolved.

b. Completeness. The standard must specify, or support specification of, all parameters within the scope of the standard that are necessary to support the development of a new implementation.

c. Ambiguity. The standard must specify required capabilities in an unambiguous way so that there is no basis for confusion as to what is required.

d. Conformance Testability. It must be feasible to test that products/implementations conform to the standard.

2-1.3 Test Procedures

a. Test Conduct

(1) Analyze Proposed Addition to the Military Standard. JITC NITFS CTE Facility personnel analyzed the proposed text of the addition to the MIL-STD-2500 and its associated standards in accordance with the general validation approach previously discussed in Section I. Each sentence, clause, and equation was evaluated for accuracy of content and absence of conflict with the portions of the standard through an exhaustive comparison of text throughout all affected standards. Additionally, the proposed standard was distributed to NTB members for review as well. All comments resulting from the documentation static review cycle were accumulated for analysis. Observations from the conduct of the implementation dynamic review (e.g. Does the implementation match the text and the desired function of nomination?) having impact on the proposed text were also evaluated. This is the primary source for assessing accuracy of content, completeness, lack of ambiguity, and testability of the proposed text.

(2) Flag Issues. Any issues uncovered were provided to the originator of the proposed text prior to detailed analysis to determine if they are substantive. This resulted in the elimination of some issues.

(3) Data Collection. Data collected included any anomalies or issues regarding the proposed text identified during the Documentation Static Review and Implementation Dynamic Review along with associated comments and recommended resolutions.

2-1.4 Results. Appendices B through G contain the final version od the Appendix I contains the results of the documentation static review which were documented in correspondence between JITC and the originator of the proposed text of the addition to the military standard. Additionally, Tables I-1 through I-3 provide a detailed listing of the comments provided.

a. Criteria Related. During static review of the proposed additions to the NITFS, seven administrative issues, twenty-nine clarifications, eight conflicts, ten occurances of missing information, and one inaccuracy were found.

b. Other. None.

2-1.5 Analysis and Discussion. All recommended changes to the proposed text addition to the standard were incorporated into the revision of the proposed text or clarified with the JITC and the test sponsor. The NITFS CTE Facility developed a test tool to conduct the implementation dynamic review portion of this validation test. All essential comments were resolved.

2-1.6 Conclusion. For those features demonstrated, the proposed text addition for tagged record extension formats for 12 bit JPEG Post Processing and Forward Error Correction addition to the military standard is technically correct, consistent, complete, and testable for its proposed application with the overall NITFS.

2-2 SUBTEST II, IMPLEMENTATION DYNAMIC REVIEW

2-2.1 Objective. To determine to what extent sample tagged record extension formats within NITF files are compliant with the proposed addition to the MIL-STD-2500 and the NITFS and to what extent a sample NITFS implementation can be adapted to use the additional tagged record extension formats.

2-2.2 Criteria. Tag fields shall be filled in accordance with appropriate tables as defined in section I and attached appendices.

a. NHECCA registered tags will include the correct FEC values and will be formatted according to Appendix C NHECCA record tagged extension description.

b. NHECCA registered tags will be located in the UDID portion of the NITF file main header.

c. NIECCA registered tags will include the correct FEC values and will be formatted according to Appendix D NIECCA record tagged extension description.

d. NIECCA registered tags will be located in the UDID portion of the NITF file image sub header.

e. NBLOCA registered tags will include the accurate index values and will be formatted according to Appendix E NBLOCA record tagged extension description, for both single and multiple blocked images.

f. NBLOCA controlled tags will be located in the IXSHD portion of the NITF file image sub header.

g. IOMAPA controlled tags will include the correct values and format according to Appendix D, IOMAPA record tagged extension description and the output process parameters selected to postprocess image data.

h. IOMAPA controlled tags will be located in the IXSHD portion of the NITF file.

i. APP_8 JPEG marker segments will include the correct minimum block values and format according to Appendix F, for both single and multiple blocked reference images.

j. APP_9 JPEG marker segment will include the correct FEC values and format according to Appendix G, for both single and multiple blocked images.

k. As a minimum, receiving implementations must successfully decompress and display an image file containing IOMAPA, NHECCA, NIECCA and NBLOCA tags, but may ignore this tag data.

l. Supporting implementations must decompress an image file, correctly utilizing the values contained in the APP_8 JPEG segment marker and the IOMAPA tag, for mapping methods 0 through 3.

m. Implementations must utilize the Error Correction Code data streams to correctly decompress NITF files containing no more than 5 bytes errors per 152 virtual message bytes in portions of the NITF file protected by ECCs.

n. Sample tagged NITF files must comply to the NITFS requirements for 12 bpp JPEG/DCT compression and decompression as specified in the NITFS Certification Test and Evaluation program Plan, JIEO Circular 9008, para 5-8.

2-2.3 Test Procedures

a. Test Conduct. The test team developed candidate test scenarios and test cases to evaluate the above list of criteria. The test team presented the sample implementation with the candidate test scenario NITF images and a selection for postprocessing type to be applied to the image. After the implementation compressed the image, the resulting NITF file was examined for accuracy and format according the proposed RFCs to MIL-STD-2500 in Appendices C through H. The sample implementation was required to process, compress, pack, and generate NITF files containing IOMAPA tagged data as well as unpack, post process, and decompress such files to an uncompressed NITF file. This uncompressed file was checked for format.

b. Data Collection. Data collection requirements for the both Interpret/Unpack and Generate/Pack tests are as follows:

(1) Annotated data collection forms with any IOMAPA related anomalies noted.

(2) Hard copies of control IOMAPA uncompressed image(s).

(3) Hard copies of the comparison between the SUT's decoder quantization values and the reference decoder values.

2-2.4 Results

a. Criterion Related. Table 2 lists Subtest 2 results. The RCJPEG Simulator has not implemented the capability to act on ECC codes to correct bit-stream errors and does not have features to use the block index values when unpacking an NITF file.

b. Other. None.

2-2.5 Analysis and Discussion

a. Criterion Related. The technical capabilities of ECC based unpack features not implemented in the RCJPEG Simulator have been previously evaluated during the validation testing of the Tactical Communications Protocol version 2 (TACO2) which uses the same ECC algorithm. Specific uses of the JPEG block indices (NBLOCA) in the proposed RFC's is not specified, nor are they intended to be specified. NBLOCA is provided for the convenience of an unpacker, not for mandated use. The new test criteria resulting from the validation effort will be nominated as proposed changes to JIEO Circular 9008, NITFS Certification Test and Evaluation Program Plan, dated 30 June 1993.

b. Other. None.

2-2.6 Conclusion. The RCJPEG Simulator is able to correctly generate 12 bit/pixel JPEG compressed image data, generation of properly formatted and populated error correction codes, and generation of post processing tags for inclusion into a version 2.0 NITF file according to the proposed RFC's. In addition, RCJPEG Simulator can unpack NITF files containing these features, but it has not implemented capabilities which exemplify the use of some of this information.

Table 1 Subtest 2 Results

Feature	Pack Correctly Pass/Fail	Unpack & Use Pass/Fail
NBLOCA Tag	Pass	NI
APP9 (APP6 Extension 0002)	Pass	NI
APP8 (App6 Extension 0001)	Pass	Pass
IOMAPA Tag, Methods 0-4	Pass	Pass
NHECCA & NIECCA Tags	Pass	NI
NI - Not Implemented in software		

SECTION III: CONCLUSIONS AND RECOMMENDATIONS

3-1 CONCLUSIONS

3-1.1 NITFS Compliance. The JITC determined that for those features demonstrated the proposed text addition to the Military Standard 2500 for 12 bit Pre and Post-Processing and Forward Error Correction (FEC) is free of conflicts, complete, unambiguous, and implementable into the NITFS. The NITFS is readily adaptable to include 12 bit Pre and Post-Processing and FEC data. Those features of the proposed text implemented in the RCJPEG Simulator have been successfully tested. The RCJPEG Simulator was able to correctly generate 12 bit/pixel JPEG compressed image data, generation of properly formatted and populated error correction codes, and generation of post processing tags for inclusion into a version 2.0 NITF file according to the proposed RFC's. In addition, RCJPEG Simulator can unpack NITF files containing these features. Tables 3-1 and 3-2 list the test results.

3-1.2 Other Conclusions. The technical capabilities of ECC based unpack features not implemented in the RCJPEG Simulator have been previously evaluated during the validation testing of the Tactical Communications Protocol version 2 (TACO2) which uses the same ECC algorithm. Specific uses of the JPEG block indices (NBLOCA) in the proposed RFC's is not specified, nor are they intended to be specified. NBLOCA is provided for the convenience of an unpacker, not for mandated use.

3-2 RECOMMENDATIONS

a. The new test criteria resulting from the validation effort be nominated as proposed changes to JIEO Circular 9008, NITFS Certification Test and Evaluation Program Plan, dated 30 June 1993.

b. The text detailing the NIECCA, IOMAPA, NBLOCA, and NHECCA tags by placed in the NITF Tag Registry.

c. The text describing the JPEG Data Application Segments; App8 and App9, be incorporated in to NITFS Mil-Std 188-198 NITF JPEG and the NITF profile to the JPEG ISO document.

TABLE 3-1 GENERATE TEST

CRITERIA/FEATURE	PASS/FAIL	REMARKS
1. NHECCA Tag:		
a. Format and correct FEC	PASSED	
b. Located in UDID	PASSED	
2. NIECCA Tag:		
a. Format and correct FEC	PASSED	
b. Located in UDID	PASSED	
3. NBLOCA Tag:		
a. Format and correct values	PASSED	
b. Located in IXSHD	PASSED	
4. IOMAP Tag:		
a. Format and correct values		
1) Method 0	PASSED	
2) Method 1	PASSED	
3) Method 2	PASSED	
4) Method 3	PASSED	
b. Located in IXSHD	PASSED	
5. APP8 marker segment has correct value and format	PASSED	
6. APP9 marker segment has correct value and format	PASSED	

Table 3-2 INTERPRET TEST

CRITERIA/FEATURE	PASS/FAIL	REMARKS
1. Interpret a file containing IOMAPA, APP8, NIECCA, NHECCA, APP9, and NBLOCA tags; however, application may ignore this tag data.	PASSED	
2. Interpret an NITF file containing IOMAPA tag and APP8 marker and utilize this embedded data.		
1) Method 0	PASSED	
2) Method 1	PASSED	
3) Method 2	PASSED	
4) Method 3	PASSED	
3. Use the NHECCA tag to display an NITF file containing errors.	NI	
4. Use the NIECCA tag to display an NITF file containing errors. . . .	NI	
5. Use the NBLOCA tag data to selectively decompress a blocked JPEG NITF file.	NI	
6. Use the APP9 marker to display an NITF file containing errors in the JPEG Frame level. . .	NI	
7. Use the APP9 marker to display an NITF file containing errors in the JPEG Scan level. . . .	NI	

APPENDIX A

GLOSSARY

ACRONYM	DEFINITION
APP	Application Segment
ARIDPCM	Adaptive Recursive Interpolated Differential Pulse Code Modulation
CDE	Controlled Data Extensions
CFS	Center for Standards
CIO	Central Imagery Office
CTE	Certification Test and Evaluation
DCT	Discrete Cosine Transform
DISA	Defense Information Systems Agency
DMA	Defense Mapping Agency
DOD	Department of Defense
DPPDB	Digital Point Positioning Data Base
ECC	Error Correction Code
FEC	Forward Error Correction
IC	Image Compression
IMODE	Image Mode
IOMAPA	CDE tag for Input/Output Amplitude Mapping Methods
ISMC	Imagery Standards Management Committee
IXSHD	Image Extended Subheader Data
JIEO	Joint Interoperability and Engineering Organization
JITC	Joint Interoperability Test Command
JPEG	Joint Photographic Experts Group
LUT	Look-Up Table
MIL-HDBK	Military Handbook
MIL-STD	Military Standard
MOA	Memoranda of Agreement
MOT	Means of Testing

APPENDIX A

GLOSSARY (Continued)

ACRONYM	DEFINITION
NBLOCA	CDE Tag for Image Frame offsets
NHECCA	Tagged Record Extension Description
NIECCA	Tagged Record Extension Description
NIMA	National Imagery and Mapping Agency
NITF	National Imagery Transmission Format
NITFS	National Imagery Transmission Format Standard
NTB	NITFS Technical Board
PIA	Profile for Imagery Architecture
RFC	Request For Change
SUT	System Under Test
TIR	Test Incident Report
UDID	User Defined Image Data
VQ	Vector Quantization

APPENDIX B

MIL-STD RFC 95-017B   DRAFT 1/2/95 VERSION B

30.2.3 IOMAPA Tagged Record Extension Description

The IOMAPA tagged extension contains the data necessary to perform the output amplitude mapping process for each scan within each image frame. This post-processing is applied after the image data has undergone the data expansion process utilizing the 12 Bit JPEG/DCT algorithm.

The output amplitude mapping function is generally the inverse of the input amplitude mapping function that is performed as a pre-processing step before the data compression process is executed.

Note: An exception to this case is when the output of the compression is scaled by a factor (S2) to change the precision of the output product relative to the input data precision.

The explanation of the input amplitude mapping is included to describe the pre-processing performed before the compression process.
The pre-processing steps are shown in Figure B3 and the post-processing steps are shown in Figure B4 for mapping methods 1 through 3.

```
   Figure B3: Input Amplitude Re-mapping Before Compression
                (Mapper Type 1 Through 3)

 -------------Input Remapping Process-----------------
 |  _______________  ______________  _______________ |  ____________   _____
 |  | Minimum     |  | Scale Data |  | Input Ampl. | |  | 12 Bit   |  |    |
===>| Value       |=>| To 12 Bits |=>| Mapping     |===>| JPEG     |=>|    |
 |  | Subtraction |  | Using S1   |  | Table       | |  | DCT COMP |  |    |
 |  _______________  ______________  _______________ |  ___________|  |    |
 |           ^             ^                ^        |                |    |
 |-----------|-------------|----------------|---------                | N  |
             |             |                |                         | I  |
             |             |                |        output mapper    | T  |
             |             |          Mapping Table ---------------->| F  |
             |             |          Specification    stored in      |    |
             |             |                            IOMAPA        | F  |
             |             |                                          | I  |
             |        S1 Scaling ------------------------------------>| L  |
             |          Factor                     stored in          | E  |
             |                                      IOMAPA            |    |
             |                                                        |    |
         Min. Value  ----------------------------------------------->|    |
         For Scan            stored in JPEG APP8                      |    |
                                                                      _____
```

Figure B4: Output Amplitude Re-mapping After Expansion
(Mapper Type 1 Through 3)

```
                    -------------Output Remapping Process-----------------
_____  ____________ |  ______________  _________________  ______________ |
|   |  | 12 Bit   | |  | Output     |  | Scale Data    |  | Scaled     | |
|   |=>| JPEG     |===>| Amplitude  |=>| To (12-S1-S2) |=>| Minimum    |===>
|   |  | DCT EXP. | |  | Mapping    |  | Bits Using    |  | Value      | |
|   |  |          | |  | Table      |  | S1 & S2       |  | Addition   | |
|   |  ___________| |  ______________  _________________  ______________ |
|   |               |         ^            ^     ^            ^    ^     |
| N |               ----------|------------|-----|------------|----|------
| I |                         |            |     |            |    |
| T |                         |            |     |            |    |
| F |------------------> Mapping Table     |     |            |    |
|   |   From IOMAPA      Specification     |     |            |    |
| F |                                      |     |            |    |
| I |                                      |     |            |    |
| L |------------------------------------> S1    |            |    |
| E |         From IOMAPA                        |            |    |
|   |                                            |            |    |
|   |-----------------------------------------> S2 --------> S2    |
|   |              From IOMAPA                                     |
|   |                                                              |
|   |---------------------------------------------------------> Min. Value
|   |                  From JPEG APP8                            For Scan
_____
```

30.2.3.1 Applicable Documents

Do we need to list the JPEG MIL-STD in regards to APP8 ?

30.2.3.2 Format Description and Mapping Method Functions

The IOMAPA data extension is used to transfer the required information needed for the inverse of the input mapping function, i.e., the output amplitude mapping function which is applied to the image data after expansion.

Tables B5 through B8 in paragraph 30.2.3.2.3 define the format for the NITF 2.0 controlled tagged record extensions bearing the tag of IOMAPA. The IOMAPA tag is meant to be stored in the image sub-header portion of the NITFS file structure. Portions of this tagged record extension are variable depending upon the value of the MAP_SELECT field within the extension.

30.2.3.2.1 Functionality of NITF JPEG/DCT Compressor When Utilizing the IOMAPA Tagged Record Extension

The input amplitude mapping function takes the image data with a known minimum value and performs a three step pre-processing function on each scan contained in the image frame before it is sent to the JPEG/DCT compressor.

The first operation subtracts the minimum pixel intensity for each scan from each pixel in the corresponding scan of the image frame. For example, the minimum value for scan 1 is subtracted from the pixels contained in the scan 1 data block. The minimum value for each scan is stored in the JPEG application segment, APP8, in order to pass this information to the expander.

The second step in the mapping process is to use the S1 factor to scale the data up to a 12 bit precision.

The third step in the mapping process is to apply an input mapping function specified as part of a compression database to the data. The compressor fills in the values of the IOMAPA extension from the compression database defining the appropriate output amplitude mapping function to be used by the expander.

In actual practice, the second and third steps can be performed with a scaled lookup table in order to gain efficiency in the implementation of the input mapping process.

If the MAP_SELECT field is equal to 0, then the subtraction of the minimum value from each block shall not be performed. The second and third steps shall also be bypassed because the mapper type 0 is used to turn off the remapping process. However, the data can be scaled with the output mapper after the JPEG expansion to decrease the precision of the data if the S2 factor is non-zero.

30.2.3.2.1.1 Input Amplitude Mapping Method 0

The amplitude mapping method 0 is used to turn off the minimum value subtraction and remapping pre-processing options. The minimum values of the scan is loaded into APP8, and a non-zero S2 output scaling factor can be loaded into the IOMAPA tag, but the data remains unchanged before it is compressed.

30.2.3.2.1.2 Input Amplitude Mapping Method 1

Table B6 describes the format of the controlled tag extension used to pass the parameters used in the amplitude mapping method 1. The controlled tag extension (method 1) contains a value by value listing or table for the output lookup process. The output lookup table is the inverse mapping of the input lookup table used by the compressor. The input mapping table is contained in a compressor database, but is not needed by the expander and not included in the IOMAPA tag. The tag also contains the input scale factor value S1, and the output precision scale change value S2.

The input amplitude mapping process which utilizes the input amplitude mapping table shall be defined as:

IXX = (IX - MIN) * ISF	Scale the original data to 12 bits after the subtraction of the minimum value
If IXX is less than 0 then IXX = 0	Clamp the value to the limits for input amplitude function.
If IXX is greater than IXMAX then IXX = IXMAX	
IXXX = input_amplitude_map_table[IXX]	Input amplitude mapping table with starting index of 0 used to remap value

where:

IX	Original Pixel Data
MIN	Minimum pixel value for image block and included in the NITF JPEG application segment APP8
ISF = 2**(S1)	Scale Factor Exponent where S1 is a data item included in IOMAPA
IXX	Scaled Original Pixel Data
IMAX = 4096	
IXMAX = IMAX - 1	Maximum Value for Input to Map Table

IXXX Re-mapped Image Pixel Data

int[] Denotes integer truncation

Note: The resultant re-mapped value shall then be clamped to ensure that it is greater than or equal to zero and less than or equal to IXMAX,

30.2.3.2.1.3 Input Amplitude Mapping Method 2

Table B7 describes the format of the controlled tag extension used to pass the parameters needed for amplitude mapping method 2. If the MAP_SELECT flag is set to 2, a generalized log mapping shall be utilized as the basis for the input amplitude mapping function. The parameters R, S1, and IMAX shall be utilized to generate the function. The parameters R, S1, and S2 shall be loaded into the IOMAPA extension.

The input amplitude mapping process for when the MAP_SELECT is set to 2 is defined below:

IX = Original Pixel Value

IXX = IX - MIN Subtract the minimum value for the image block

If R is not equal to 1.0 Perform log mapping

IXXX = int[(B * ln(1.0 + A*IXX)) + 0.5]

else

IXXX = IXX * ISF Special case for log mapping if R = 1.0

where:

IX Original Image Pixel Data

IXX Image pixel data after the minimum value subtraction

MIN Minimum pixel value for image block and included in the NITF JPEG application segment APP8

A = (R-1.0)/IXMID

B = IXMAX/(ln(1.0 + A*ISMAX))

R	Log Ratio data item loaded into IOMAPA
IMAX = 4096	IMAX shall be 4096 for 12 Bit JPEG/DCT
IXMAX = IMAX - 1	
ISMAX = (IMAX/ISF)-1	Scaled maximum
IXMID = (IMAX/(2*ISF)	Scaled mid-point
ISF = 2**(S1)	Scale Factor S1
IXXX	Re-mapped Image Pixel Data
int[]	Denotes integer truncation

Note: The resultant re-mapped value IXXX shall then be clamped to ensure that it is greater or equal to zero and less than or equal to IXMAX.

30.2.3.2.1.4 Input Amplitude Mapping Method 3

Table B8 describes the format of the controlled tag extension used to pass the parameters for amplitude mapping method 3. Mapping method 3 uses a 3 segment polynomial mapping process where each interval is described by a fifth order polynomial. The starting point for each interval and a set of six coefficients defining the polynomial for each segment shall be database items. The coefficients stored in the IOMAPA tag are different from the ones used in the input mapping process. The coefficients stored in the tag usually reflect the inverse mapping characteristics of the input mapper coefficients. The input mapper coefficients are stored in a compressor database, but are not needed by the expander and are not included in the IOMAPA tag.

If the MAP_SELECT flag is set to the value 3, the following segmented polynomial mapping shall be utilized for each pixel before the 12 Bit JPEG/DCT compression process.

$IX = $ Original Pixel Value

$IXX = IX - MIN$	Subtract the minimum value for the image block
$IXXX = IXX * ISF$	Shifted input value IXX is scaled by ISF

The scaled original pixel (IXXX) value shall determine which segment of the polynomial function shall be utilized.

Segment (J) shall be defined as

$$X(J-1) <= IXXX < X(J) \quad \text{For } J = 1, 2, \text{ and } 3$$

where

X(J) are segment bounds

$X(0) = 0$ and $X(3) = 4096$

X(1) and X(2) shall be included in the NITF CDE IOMAPA

The scaled input pixel value (IXXX) shall be mapped using the coefficients (ai) for the appropriate polynomial segment as defined above. These coefficients are stored in a compressor database and are not included in the IOMAPA tag. Output coefficients, which perform the inverse mapping operation of the (ai)'s, are included in the IOMAPA tag. The input mapping expression for the polynomial function is given below:

$$IZ = IXXX - X(J-1)$$

$$IY = \mathrm{int}[a0 + a1*IZ + a2*(IZ**2) + a3*(IZ**3) + a4*(IZ**4) + a5*(IZ**5) + 0.5]$$

where:

IX	Original pixel value
IXX	Image pixel data after the minimum value subtraction
MIN	Minimum pixel value for image block and included in the NITF JPEG application segment APP8
IXXX	Scaled value to determine segment number
$ISF = 2**(S1)$	Scale Factor (S1 from IOMAPA)

a0, a1, a2, ..., a5	6 Input Mapper Coefficients For Segment J { X(J-1) < = IXXX < X(J) }
X(J-1)	Lower Boundary For Segment J
IY	Re-mapped image pixel value

IMAX = 4096

IXMAX = IMAX - 1 = 4095

int[]	Denotes integer truncation

Note: The output of the polynomial mapping function (IY) shall be clamped to ensure that it is greater than or equal to zero and less or equal to IXMAX.

30.2.3.2.2 Functionality of NITF JPEG/DCT Expander When Utilizing the IOMAPA Tagged Record Extension

The output amplitude mapping function takes the reconstructed image data from the JPEG expansion process and performs a three step post-processing function on the data unless mapping method 0 is applied. The first step in the remapping process is to apply an output mapping function specified by the IOMAPA extension present in the NITF file. The second operation rescales the data values using the S1 and S2 values. The final operation adds the minimum value extracted from the JPEG APP8 application segment to each pixel value.

If the MAP_SELECT field is equal to 0, then the remapping amplitude function and the addition of the minimum value shall not be performed. Only the S2 factor shall be used to change the precision of the data to (orig_precision-S2) bits.

30.2.3.2.2.1 Output Amplitude Mapping Method 0

The amplitude mapping method 0 code describes to the interpreter of the NITF file that no input or output remapping function or minimum value shift is applied to the data. However, if the S2 field is not equal to zero, the data values shall be scaled by the factor of 2**S2. The output scaled pixel value shall use the following expression:

OX = int[(IY/OSF) + 0.5]

where:

IY = Pixel Value From JPEG Expander

OSF = 2**(S2)

OX = Output Precision Scaled Pixel Value

30.2.3.2.2.2 Output Amplitude Mapping Method 1

Table B6 describes the format of the controlled tag extension for amplitude mapping method 1.

The IOMAPA tag (method 1) contains a value by value listing or table for the output lookup process. The tag also contains the input scale factor value S1, and the output precision scale change value S2.

The output amplitude mapping process which utilizes the output amplitude mapping table shall be defined as:

If IY is less than 0 then IY = 0	Clamp the input to the output amplitude function.
If IY is greater than IXMAX then IY = IXMAX	
IXX = output_amplitude_map_table[IY]	Virtual array with the values of the output amplitude mapping table loaded starting at index 0.
OX = int[(IXX/(ISF*OSF)) + 0.5] + int[(MIN/OSF)+0.5]	Scaled Output Data with scaled image block minimum added.

where:

IY	Pixel Data From JPEG Expander
IMAX = 4096	
IXMAX = IMAX - 1	Maximum Value for Input to Map Table
ISF = 2**(S1)	Scale Factor Exponent where is a data item included in IOMAPA
OX	Re-scaled Image Pixel Data
OSF = 2**(S2)	Scale Factor Exponent where S2 is a data item included in IOMAPA

MIN	Minimum pixel value for image block extracted from the NITF JPEG application segment APP8
OMAX = (IMAX/(ISF*OSF))-1	Maximum Value Clamp for Final Output
int[]	Denotes integer truncation

Note: The resultant output shall then be clamped to ensure that it is greater than or equal to zero and less than or equal to OMAX.

30.2.3.2.2.3 Output Amplitude Mapping Method 2

Table B7 describes the format of the controlled tag extension for amplitude mapping method 2.
`

If the MAP_SELECT flag is set to 2, a generalized log mapping shall be utilized as the basis for the output amplitude mapping function. The parameters R, S1, S2, and IMAX shall be utilized to generate the function.

The parameters R, S1, and S2 shall be extracted from the IOMAPA tag.

The output amplitude mapping process for when the MAP_SELECT set to 2 is defined below:

If IY is less than 0 then IY = 0	Clamp the input to the function.

If IY is greater than IXMAX then
  IY = IXMAX

If R is not equal to 1.0
  IXX = int[((exp(IY/B)-1.0)/A)/OSF + 0.5]
else
  IXX = int[(IY/(ISF*OSF)) + 0.5]

IX = int[(IXX/OSF) + 0.5]	Scale Resultant Output Data
OX = IX + int[(MIN/OSF) + 0.5]	Scaled Output Data with Scaled Image Block Minimum Added

where:

IY	Clamped Pixel Data From JPEG Expander

R	Log Ratio data item from IOMAPA Tag
A = (R-1.0)/IXMID	
B = IXMAX/(ln(1.0 + A*ISMAX))	
IXMID = (IMAX/(2*ISF)	Scaled mid-point
ISMAX = (IMAX/ISF)-1	Scaled maximum
IMAX = 4096	IMAX shall be 4096 for 12 Bit JPEG/DCT
IXMAX = IMAX - 1	Maximum input
ISF = 2**(S1)	Scale Factor (S1 from IOMAPA)
OSF = 2**(S2)	Scale Factor (S2 from IOMAPA)
IX	Re-scaled output mapped pixel (with minimum still subtracted)
MIN	Minimum pixel value for image block and included in the NITF JPEG application segment APP8
OX	Re-scaled Image Pixel Data
OMAX = (IMAX/(ISF*OSF))-1	Maximum Value for Final Output
int[]	Denotes integer truncation
exp()	Exponential Function (e**x)

Note: The resultant output shall then be clamped to ensure that it is greater than or equal to zero and less than or equal to OMAX.

30.2.3.2.2.4 Output Amplitude Mapping Method 3

Table B8 describes the format of the controlled tag extension for amplitude mapping method 3. Mapping method 3 uses a 3 segment polynomial mapping process where each interval is described by a fifth order polynomial. The break point for each interval and a set of six coefficients defining the polynomial for each segment shall be extracted from the IOMAPA tag.

If the MAP_SELECT flag is set to the value 3, the following segmented polynomial mapping shall be utilized for each pixel output from the expansion process.

The output pixel (IY) from the JPEG/DCT expansion process shall determine which segment of the polynomial function shall be utilized.

Segment (J) shall be defined as

X(J-1) < = IY < X(J)   For J = 1, 2, and 3

where

X(J) are segment bounds
X(0) = 0 and X(3) = 4096
X(1) and X(2) shall be included in the NITF CDE IOMAPA

The output pixel value (IY) shall be mapped using the coefficients (bi) for the appropriate polynomial segment as defined above. The expression for the polynomial function is given below:

If IY is greater than 4095, then IY = 4095.
If IY is less than 0, then IY = 0.

IZ = IY - X(J-1)

IXX = int[b0 + b1*IZ + b2*(IZ**2) + b3*(IZ**3)
          + b4*(IZ**4) + b5*(IZ**5) + 0.5]

Where the coefficients b0 through b5 are included in the NITF CDE IOMAPA.

The output of the polynomial mapping function (IXX) shall be scaled by the following relationship:

IX = int[(IXX/(ISF*OSF)) + 0.5]

OX = IX + int[(MIN/OSF) + 0.5]

where:

IY	Pixel value from expansion process (Determines Segment Number Location)
X(J-1)	Lower Boundary For Segment J
b0, b1, b2, ..., b5	6 Output Mapper Coefficients For Segment J { X(J-1) < = IY < X(J) }
IXX	Intermediate value from polynomial equation
IX	Re-scaled output mapped pixel (with minimum still subtracted)
ISF = 2**(S1)	Scale Factor (S1 from IOMAPA)
OSF = 2**(S2)	Scale Factor (S2 from IOMAPA)
OX	Re-scaled Image Pixel Data
MIN	Minimum pixel value for image block and extracted from the NITF JPEG application segment APP8
OMAX = ((IMAX/(ISF*OSF)) - 1)	Final output value clamp
IMAX = 4096	
int[]	Denotes integer truncation

The resultant output (OX) shall then be clamped to ensure that it is greater than or equal to zero and less than or equal to OMAX.

30.2.3.2.3 IOMAPA Tagged Record Extension Format Tables

Table B-5: IOMAPA Format for Mapping Method 0

(R) = required, (O) = Optional, and (C) = Conditional

Field	Description	Length (bytes)	Value Range	Type
CETAG	Unique Extension Identifier	6	IOMAPA	R
CEL	Length of CEDATA	5	00006	R

	Fields			
BAND_ NUMBER	Band Identifier (Band = 000 for Monochrome or Single Band Imagery)	3	000-999	R
MAP_ SELECT	Mapping Method to Apply	1	0	R
S2	Scale Factor 2	2	00-11	R

Table B-6: IOMAPA Format for Mapping Method 1

(R) = required, (O) = Optional, and (C) = Conditional

Field	Description	Length (bytes)	Value Range	Type
CETAG	Unique Extension Identifier	6	IOMAPA	R
CEL	Length of CEDATA Fields	5	08202	R
BAND_ NUMBER	Band Identifier (Band = 000 for Monochrome or Single Band Imagery)	3	000-999	R
MAP_ SELECT	Mapping Method to Apply	1	1	R
TABLE_ ID	I/O TABLE USED (See note 2)	2	00-99	O
S1	Scale Factor 1 (See note 3)	2	00-11	R
S2	Scale Factor 2 (See note 4)	2	00-11	R
INPUT MAP VALUE 0	First Input Mapping Value	2	(See note 1)	R
....	....	..	....	.....
INPUT MAP VALUE 4095	Last Input Mapping Value	2	(See note 1)	R

Notes:

1. Value is stored in 2 byte unsigned integer format (Most Sign. Byte First). The binary value is limited to be greater than or equal to 0 and less than or equal 4095.

2. Table_ID is not needed to perform the output mapping function. It is used for diagnostic purposes and can be considered an optional field.

3. The value of S1 is used to scale the input data precision up to 12 bits. For the example of 8 bit input data, the S1 value would be 4.

4. The value of S2 is limited to the range where S2 < (12 - S1). Otherwise, all of the data bits would be destroyed.

Table B-7: IOMAPA Format for Mapping Method 2

(R) = required, (O) = Optional, and (C) = Conditional

Field	Description	Length (bytes)	Value Range	Type
CETAG	Unique Extension Identifier	6	IOMAPA	R
CEL	Length of CEDATA	5	00016	R
BAND_ NUMBER	Band Identifier (Band = 000 for Monochrome or Single Band Imagery)	3	000-999	R
MAP_ SELECT	Mapping Method to Apply	1	2	R
TABLE_ ID	I/O TABLE ID (See note 1)	2	00-99	O
S1	Scale Factor 1 (See note 2)	2	00-11	R
S2	Scale Factor 2 (See note 3)	2	00-11	R
R_ WHOLE	R Scaling Factor- Whole Part (See note 4)	3	000-999	R
R_ FRACTION	R Scaling Factor- Fractional Part (See note 4)	3	000-255	R

Notes:
1. Table_ID is not needed to perform the output mapping function. It is used for diagnostic purposes and can be considered an optional field.

2. The value of S1 is used to scale the input data precision up to 12 bits. For the example of 8 bit input data, the S1 value would be 4.

3. The value of S2 is limited to the range where S2 < (12 - S1). Otherwise, all of the data bits would be destroyed.

4. The R values contains two parts, the fractional part and the whole part. The resultant of R is derived by the expression:

R = R_WHOLE + (R_FRACTION/256)

Table B-8: IOMAPA Format for Mapping Method 3

(R) = required, (O) = Optional, and (C) = Conditional

Field	Description	Length (bytes)	Value Range	Type
CETAG	Unique Extension Identifier	6	IOMAPA	R
CEL	Length of CEDATA Fields	5	00091	R
BAND_ NUMBER	Band Identifier (Band = 000 for Monochrome or Single Band Imagery)	3	000-999	R
MAP_SELECT	Mapping Method to Apply	1	3	R
TABLE_ID	I/O TABLE ID (See note 1)	2	00-99	O
S1	Scale Factor 1 (See note 2)	2	00-11	R
S2	Scale Factor 2 (See note 3)	2	00-11	R
NO_OF_ SEGMENTS	Number of Segments	1	3	R
X_1	Segment Boundary 1	4	0000-4095	R

X_2	Segment Boundary 2	4	0000-4095	R
OUT_B0_1	B0 Coefficient of 1st Segment	4	(See note 4)	R
OUT_B1_1	B1 Coefficient of 1st Segment	4	(See note 4)	R
OUT_B2_1	B2 Coefficient of 1st Segment	4	(See note 4)	R
OUT_B3_1	B3 Coefficient of 1st Segment	4	(See note 4)	R
OUT_B4_1	B4 Coefficient of 1st Segment	4	(See note 4)	R
OUT_B5_1	B5 Coefficient of 1st Segment	4	(See note 4)	R
OUT_B0_2	B0 Coefficient of 2nd Segment	4	(See note 4)	R
OUT_B1_2	B1 Coefficient of 2nd Segment	4	(See note 4)	R
OUT_B2_2	B2 Coefficient of 2nd Segment	4	(See note 4)	R
OUT_B3_2	B3 Coefficient of 2nd Segment	4	(See note 4)	R
OUT_B4_2	B4 Coefficient of 2nd Segment	4	(See note 4)	R
OUT_B5_2	B5 Coefficient of 2nd Segment	4	(See note 4)	R
OUT_B0_3	B0 Coefficient of 3rd Segment	4	(See note 4)	R
OUT_B1_3	B1 Coefficient of 3rd Segment	4	(See note 4)	R
OUT_B2_3	B2 Coefficient of 3rd Segment	4	(See note 4)	R
OUT_B3_3	B3 Coefficient of 3rd Segment	4	(See note 4)	R
OUT_B4_3	B4 Coefficient of 3rd Segment	4	(See note 4)	R
OUT_B5_3	B5 Coefficient of 3rd Segment	4	(See note 4)	R

Notes:

1. Table_ID is not needed to perform the output mapping function. It is used for diagnostic purposes and can be considered an optional field.

2. The value of S1 is used to scale the input data precision up to 12 bits.

3. The value of S2 is limited to the range where S2 $<$ (12 - S1). Otherwise, all of the data bits would be destroyed.

4. The value is stored in 4 byte IEEE single precision floating point format. Value range is the range available in the standardized 4 byte IEEE single precision floating point format.

APPENDIX C

NHECCA TAGGED RECORD EXTENSION 10/5/94

30.2.4 NHECCA Tagged Record Extension Description

The Forward Error Correction (FEC) codes stored in this extension provides acceptable protection to the NITF header from random bit errors in the range of one in 10^6 bits.

Table B9 defines the format for the registered tagged record extension to the NITF 2.0 bearing tag NHECCA. This extension contains FEC codes which result from FEC coding to the NITF header (including the registered and controlled extensions) with the exception of the NHECCA tagged record extension.

Table B9 NITF Registered Data Extension For NITF Header Level ECC

FIELD NAME	BYTES	DESCRIPTIO N	FORMAT ASCII	UNITS	RANGE
RETAG	6	UNIQUE EXTENSION IDENTIFIER	NHECCA	N/A	N/A
REL	5	LENGTH OF ENTIRE RECORD	(See note 1)	BYTES	0009 = 99988
ECC TYPE	1	TYPE OF ECC	1	N/A	0-No ECC; 1-RS-ECC; 2-BCH ECC
ECC BEGIN SYNC	8	SYNC CODE FIELD FOR NITF HEADER ECC	HDRBSYNC	N/A	N/A
ECC DATA	(See note 2)	ECC CODES	N/A	N/A	ANY BINARY VALUES

Notes: 1. This value is a computed quantity based upon the number of ECC codes resulting from applying the ECC algorithm to the NITF image header. This value includes additional data extension bytes.

2. This value is a computed quantity based upon the number of ECC codes resulting from applying the ECC algorithm to the NITF image header.

30.2.4.1 Applicable Documents

MIL-STD-2045-44500, Tactical Communications Protocol 2 (TACO2), 18 June 1993 and Notice 1, 29 July 1994.

30.2.4.2 Format Description

The FEC code utilized to error correct the NITF header and registered tag record extensions is based upon MIL-STD-2045-44500 (TACO2), paragraph 5.4.2.1, and is termed FEC-1 code.

For purposes of applying the FEC-1 code, the NITF header and registered tag extensions shall be logically separated into virtual data grams of 152 bytes in length.

If the applicable NITF data does not break evenly into whole virtual datagrams, the remaining data, i.e., the last virtual datagram shall have fewer than 152 message bytes, but will still result in a 10 byte FEC.

The code is applied to the 152 message byte virtual datagrams comprising the NITF overhead data.

The resultant Reed-Solomon check bytes (10 bytes per 152 message bytes) are stored separately in the NHECCA tagged record extension. ***The storing of the check bytes separately from the datagrams is different than what paragraph 5.4.2.1 of the FEC-1 code describes.***

The resulting length of the NHECCA field will vary dependent upon the length of the NITF header (excluding the NHECCA tag extension field) as the number of FEC codes directly affect the length of the NHECCA field.

APPENDIX D

NIECCA TAGGED RECORD EXTENSION 10/5/94

30.2.5 NIECCA Tagged Record Extension Description

The Forward Error Correction (FEC) codes stored in this extension provides acceptable protection to the NITF header from random bit errors in the range of one in 10^6 bits.

Table B10 defines the format for the registered tagged record extension to the NITF 2.0 bearing tag NIECCA. This extension contains FEC codes which result from FEC coding to the NITF header (including the registered and controlled extensions) with the exception of the NIECCA tagged record extension.

Table B10 NITF Registered Data Extension For NITF Image Sub-Header Level ECC

FIELD NAME	BYTES	DESCRIPTIO N	FORMAT ASCII	UNITS	RANGE
RETAG	6	UNIQUE EXTENSION IDENTIFIER	NIECCA	N/A	N/A
REL	5	LENGTH OF ENTIRE RECORD	(See note 1)	BYTES	00009 = 99988
ECC TYPE	1	TYPE OF ECC	1	N/A	0-No ECC; 1-RS-ECC; 2-BCH ECC
ECC BEGIN SYNC	8	SYNC CODE FIELD FOR NITF HEADER ECC	SUBHDRBS	N/A	N/A
ECC DATA	(See note 2)	ECC CODES	N/A	N/A	ANY BINARY VALUES

Notes:
1. This value is a computed quantity based upon the number of ECC codes resulting from applying the ECC algorithm to the NITF image subheader. This value includes additional data extension bytes.
2. This value is a computed quantity based upon the number of ECC codes resulting from applying the ECC algorithm to the NITF image subheader.

30.2.5.1 Applicable Documents

MIL-STD-2045-44500, Tactical Communications Protocol 2 (TACO2), 18 June 1993 and Notice 1, 29 July 1994.

30.2.5.2 Format Description

The FEC code utilized to error correct the NITF image subheader and registered tag record extensions is based upon MIL-STD-2045-44500 (TACO2), paragraph 5.4.2.1, and is termed FEC-1 code.

For purposes of applying the FEC-1 code, the NITF image subheader and registered tag extensions shall be logically separated into virtual data grams of 152 bytes in length.

If the applicable NITF data does not break evenly into whole virtual datagrams, the remaining data, i.e., the last virtual datagram shall have fewer than 152 message bytes, but will still result in a 10 byte FEC.

The code is applied to the 152 message byte virtual datagrams comprising the NITF overhead data.

The resultant Reed-Solomon check bytes (10 bytes per 152 message bytes) are stored separately in the NIECCA tagged record extension. ***The storing of the check bytes separately from the datagrams is different than what paragraph 5.4.2.1 of the FEC-1 code describes.***

The resulting length of the NIECCA field will vary dependent upon the length of the NITF header (excluding the NIECCA tag extension field) as the number of FEC codes directly affect the length of the NIECCA field.

APPENDIX E

MIL-STD RFC 95-014B DRAFT 1/2/95 VERSION B

30.2.6 NBLOCA Tagged Record Extension

This tagged extension stores the offset of each image frame relative to each other within a NITF image. The first image frame offset is the number of bytes in the image sub-header. All of the other offsets are the number of bytes in the previous image block or frame.

This extension allows the NITF image to be accessed in a random or parallel fashion by providing the ability to find the offset to the location of the first data byte of any frame or block. This offset is determined by summing the offset values for the previous blocks, and allows direct access to a frame without reading through any portion of the image frames. For JPEG applications, these offsets are to the Start Of Image (SOI) markers which are always the first element for each JPEG
compressed frame.

Table B-11 defines the format for the NITF 2.0 controlled tagged record extension bearing the tag of NBLOCA. This extension is meant to be stored in the NITF image sub-header portion of the NITF file structure.

Table B-11: NBLOCA Format

(R) = required, (O) = Optional, and (C) = Conditional

Field	Description	Length (bytes)	Value Range	Type
CETAG	Unique Extension Identifier	6	NBLOCA	R
CEL	Length of CEDATA Fields (See Note 1)	5	00008-99988	R
FRAME_1_ OFFSET	First Image Frame Offset From Beginning of NITF Image Sub-header (See Note 2)	4	(See Note 2)	R

NUMBER_ OF_FRAMES	Number of Blocks For Which Offsets Are Listed	4	(See Note 3)	R
FRAME_2_ OFFSET	Offset in Bytes of the Beginning of the 2nd Image Frame From the Beginning of the 1st Image Frame (see note 5)	4	(see note 4)	C
....	....	....	....	....
FRAME_N_ OFFSET	Offset in Bytes of the Beginning of the Nth Image Frame From the Beginning of the N-1 Image Frame	4	(see note 4)	C

Notes:

1) This value is dependent upon the number of image frame offsets which are stored in this controlled data extension.

2) Value is stored in 4 byte unsigned binary integer representation with a range of 439 to 999999 (Bounds For Image Subheader Size). This offset is equal to the size of the image subheader.

3) Value is stored in 4 byte unsigned binary integer representation with a range of 1 to 24996 (Limits due to max size of CETAG).

4) Value is stored in 4 byte unsigned binary integer representation with a range of 1 to (2**32 - 1).

5) For JPEG applications, this is the offset between the SOI marker of the 2nd Image Frame from the SOI marker of the 1st Image Frame.

APPENDIX F

MIL-STD RFC 95-016B   DRAFT 1/2/95 VERSION B

MODIFY THE PROBLEM DESCRIPTION ON THE COVER PAGE

PROBLEM DESCRIPTION

This change provides for a minimum pixel value storage mechanism for each scan in the JPEG DCT stream.   -------

5.2.3.3.5.5.3 NITF APP8 Image Block Minimum Value

The NITF APP8 application data segment contains the minimum value for each scan of an original uncompressed image block before any preprocessing or compression steps are performed. This application segment also stores the image block index values which specify the relative image block row and image block column position of the frame.

When the NITF tag for the amplitude re-mapping process is used, (IOMAPA), the minimum values stored in APP8 are utilized by the amplitude re-mapping process described in paragraph 30.2.3 of MIL-STD-2500A. When the NITF amplitude re-mapping tag is used (IOMAPA), an APP8 application data segment must exist for each image block or frame compressed with the 12-bit JPEG algorithm. When using the JPEG DCT baseline sequential transmission mode with
monochrome imagery, the Nscans field shall be fixed at 1.

Table XVII contains the format for the APP8 segment.

Table XVII NITF JPEG APP8 Segment Format for Image Block Minimum Values

Offset	Field Value	Field Name	Length	Comments
0	0xFFE8	APP8	2	NITF APP Data Marker
2	Variable	Lp	2	Seg Length (See Note 5)
4	Generated (see note 1)	Image Block Row No.	4	(See Note 3)
8	Generated (see note 1)	Image Block Col No.	4	(See Note 3)
12	Generated	Nscan	1	Number of Scans per

	(see note 4)			Frame
13	Generated (see note 2)	Min_Value_1	2	Min. value of Scan #1 in Image Block
...	...	...	...	...
C	Generated (see note 2)	Min_Value_NScan	2	Min. value of Scan #Nscan in Image Block
C	0	Flags	2	Reserved for Future Use (see note 6)

Notes:

1) Value is 4 byte unsigned binary integer representation

2) Value is 2 byte unsigned binary integer representation

3) Image block index relative to the transmitted image. The top left image block is indexed (row, column) -> (1,1)

4) Value is 1 byte unsigned binary integer representation

5) Length Lp = 13 + (2*NScan)

6) The offset label of C is used for the conditional offsets dependent on the value of the NScan field.

Figures XVIII and XIX show the location of the APP8 application data segments relative to other NITF components.

FIGURE XVIII -- NITF 12 Bit JPEG/DCT Multiple Block File Structure (IMODE = B or P)

NO CHANGE TO FIGURE

FIGURE XIX -- NITF 12 Bit JPEG/DCT Multiple Block File Structure (IMODE = S)

NO CHANGE TO FIGURE

APPENDIX G

MIL-STD RFC 95-015B   DRAFT 1/10/96 VERSION B

5.2.3.3.5.5.4 NITF APP9 FEC (Forward Error Correction Code)

The NITF APP9 application data segment contains the FEC (Forward Error Correction) codes which are used to protect the NITF/JPEG header and misc. table data from bit errors.

The FEC codes are applied to:

1. NITF/JPEG Frame Header and Misc. Tables
2. NITF/JPEG Scan Header and Misc. Tables

Two different forms of the APP9 application data segment shall be used for each image block, one for the frame overhead data and one for the scan overhead data. The two forms are conditional forms based the value of the APP9 Type field which discriminates the frame and scan forms of the extension. The two forms are very similar with the exception of the values contained in the APP9 Type field and the 8 byte ASCII formatted SYNC Code inserted before the Reed-Solomon FEC bytes. The placement of the APP9 application segments is shown in Figures XX and XXI.

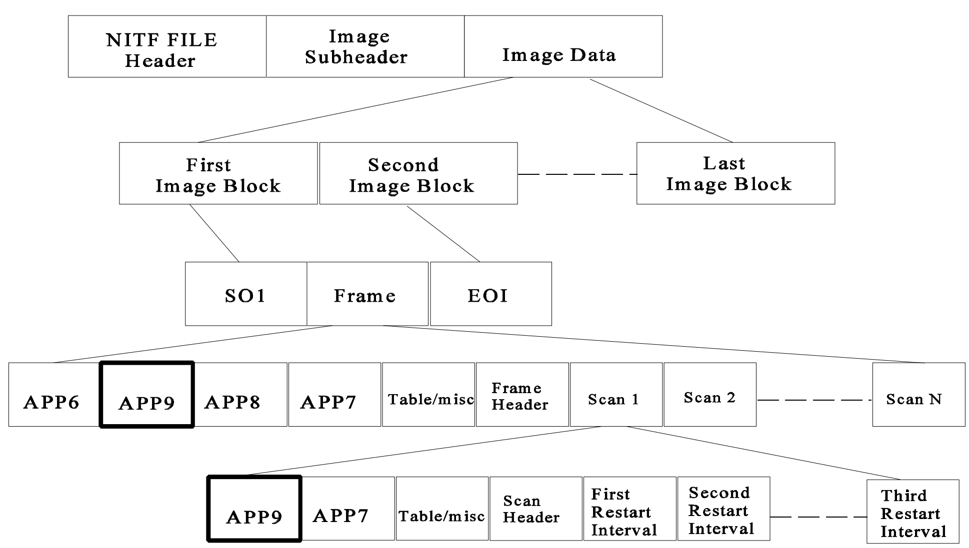

Figure XX
NITF 12BIT JPEG/DCT MULTIPLE BLOCK FILE STRUCTURE WITH FEC
(IMODE = B OR P)

NO change to FIGURE except for possibly the ORDERING of APPs following the SOI marker.

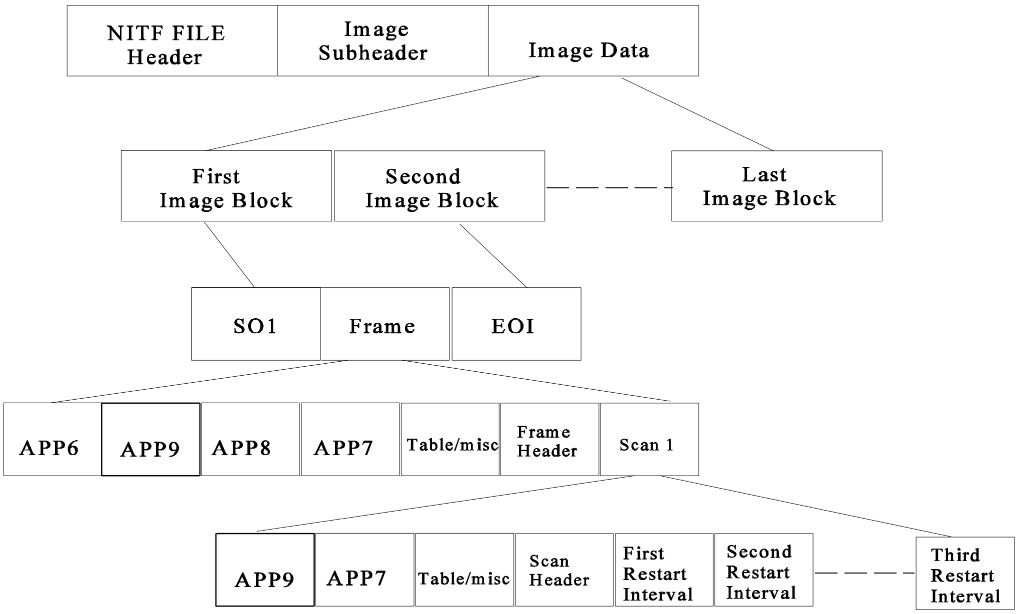

FIGURE XXI
NITF 12 BIT JPEG/DCT MULTIPLE BLOCK FILE STRUCTURE WITH FEC
(IMODE = S)

NO change to FIGURE except for possibly the ORDERING of APPs following the SOI marker.

The FEC code utilized to error protect the JPEG header and misc. tables is based upon the MIL-STD-2045-44500 (TACO2), paragraph 5.4.2.1, and is termed the FEC-1 code. For purposes of applying the FEC-1 code, the JPEG header and tables shall be logically separated into virtual datagrams of 152 bytes or less in length. For example, to build the datagrams for the Frame level APP9, the bytes from the SOI marker to the last byte of the Start of Frame (SOF) marker segment shall be used to fill the datagram stream. This includes any other allowed JPEG marker segments or APPs, other than APP9, in the tables/misc. area before the SOF marker. The end of the SOF marker segment is the end of the Frame portion of the overhead for the image block. If the number of bytes in the virtual datagram stream is less than or equal to 152 bytes there will be only 1 datagram. Otherwise, the datagram stream will be divided into 152 byte sections until the last datagram is less than or equal to 152 bytes. Each datagram section of 1 to 152 bytes will produce a 10 byte FEC code from the Reed-Solomon algorithm.

To build the datagram stream for the Scan level APP9, the bytes following the last byte of the SOF marker segment to the last byte of the Start of Scan (SOS) marker segment shall be used. This includes any other allowed JPEG marker segments or APPs, other than APP9, in the tables/misc. area before the SOS marker. The end of the SOS marker segment is the end of the Scan portion of the overhead for the image block. The datagram stream for the Scan overhead is sectioned identically to the Frame overhead stream.

Unlike the TACO2 FEC-1 protocol, the 10 byte FEC codes from all of the datagrams are concatenated within the APP9 tag, and are not interleaved within the data stream as stated in paragraph 5.4.2.1 of the FEC-1 code in MIL-STD-2045-44500 (TACO2).

The resulting length of the APP9 application segment will vary dependent upon the number of 10 byte FEC codes in the APP9 field, which is dependent on the length of the JPEG marker segments and misc. tables present in the overhead.

If the ECC type is set to zero, only the block sizes, SOI offsets, and ASCII SYNC codes are included in the APP9 extension, and Lp will have a fixed length of 24 bytes.

Tables XXII and XXIII contain the formats for the APP9 segment.

Table XXII NITF JPEG APP9 Segment Format for NITF JPEG Frame level Overhead

Offset	Field Value	Field Name	Length	Comments
0	0xFFE9	APP9	2	NITF APP Data Marker
2	Variable	Lp	2	Segment Length (See Note 1)
4	0x02	APP9 Type	1	Type Code 02 specifies Frame APP9 Extension
5	0x00 or 0x01	ECC Type	1	ECC Type Code specifies the ECC Protection Algorithm

Currently Defined Values:

0 = No ECC Codes, only offsets and SYNC Code included
1 = RS ECC Codes, SYNC Code, and offsets included

Offset	Field Value	Field Name	Length	Comments
6	(See note 2)	Offset From SOI	4	Offset in bytes from this APP9 Marker to the SOI Marker of this block
10	(See note 2)	Previous Block Size	4	Number of bytes in previous image block (See note 3)
14	(See note 2)	Current Block Size	4	Number of bytes in this image block (See note 3)
18	0x46454343 0x5353594E	Start ECC SYNC Code	8	Sync Code for Start of ECC codes (ASCII: FECCSSYN)
26	Generated (See note 4)	First 10 byte Code	10	10 Byte ECC Code from First Datagram Block
...	...	...	...	...
C	Generated (See note 5)	Last 10 byte Code	10	10 Byte ECC Code from Last Datagram Block
	(See note 6)			

Notes:

1) Length Lp = 24 + (10 * (Number_of_datagrams)) If ECC Type = 0, the Number_of_datagrams = 0.

2) Value is unsigned binary integer format of appropriate length

3) Size in bytes for the image blocks can be calculated by the adding 1 to the offset between the EOI and the SOI markers for each image block. Sizes stored in binary integer format.

4) Each datagram of 152 bytes or less produces a 10 byte ECC code using the RS FEC protection algorithm.

5) The last datagram of 152 bytes or less produces a 10 byte ECC code using the RS FEC protection algorithm.

6) The offset label of C is used for the conditional offsets dependent on the number of ECC code bytes.

Table XXIII NITF JPEG APP9 Segment Format for NITF JPEG Scan level Overhead

Offset	Field Value	Field Name	Length	Comments
0	0xFFE9	APP9	2	NITF APP Data Marker
2	Variable	Lp	2	Segment Length (See Note 1)
4	0x03	APP9 Type	1	Type Code 03 specifies Scan APP9 Extension
5	0x00 or 0x01	ECC Type	1	ECC Type Code specifies the ECC Protection Algorithm

Currently Defined Values:

0 = No ECC Codes, only offsets and SYNC Code included
1 = RS ECC Codes, SYNC Code, and offsets included

Offset	Field Value	Field Name	Length	Comments
6	(See note 2)	Offset From SOI	4	Offset in bytes from this APP9 Marker to the SOI Marker of this block
10	(See note 2)	Previous Block Size	4	Number of bytes in previous image block (See note 3)
14	(See note 2)	Current Block Size	4	Number of bytes in this image block (See note 3)
18	0x53454343 0x5353594E	Start ECC SYNC Code	8	Sync Code for Start of ECC codes (ASCII: SECCSSYN)
26	Generated (See note 4)	First 10 byte Code	10	10 Byte ECC Code from First Datagram Block

...	...	...	...	...
C	Generated (See note 5) (See note 6)	Last 10 byte Code	10	10 Byte ECC Code from Last Datagram Block

Notes:

1) Length Lp = 24 + (10 * (Number_of_datagrams)) If ECC Type = 0, the Number_of_datagrams = 0.

2) Value is unsigned binary integer format of appropriate length

3) Size in bytes for the image blocks can be calculated by the adding 1 to the offset between the EOI and the SOI markers for each image block. Sizes stored in binary integer format.

4) Each datagram of 152 bytes or less produces a 10 byte ECC code using the RS FEC protection algorithm.

5) The last datagram of 152 bytes or less produces a 10 byte ECC code using the RS FEC protection algorithm.

6) The offset label of C is used for the conditional offsets dependent on the number of ECC code bytes.

APPENDIX H

REFERENCES

The following list of references will be used in the evaluation of the proposed addition to the MIL-STD-2500.

MIL-HDBK-1300A, National Imagery Transmission Format Standard (NITFS) Handbook, 30 June 1993.

MIL-STD-2500A, National Imagery Transmission Format Version 2.0, 12 October 1994.

MIL-STD-188-198A, Joint Photographic Experts Group (JPEG) for NITFS, 12 October 1994.

MIL-STD-2045-44500, Tactical Communications Protocol 2 (TACO2), 30 June 1993 and Notice, 29 July 1994.

MIL-STD-2301, Computer Graphics Metafile (CGM) for NITFS, 30 June 1993 and Notice 1, 12 October 1994.

MIL-STD-188-196, Bi-Level Image Compression, 30 June 1993.

MIL-STD-188-197A, Adaptive Recursive Interpolated Differential Pulse Code Modulation (ARIDPCM) Compression Algorithm, 12 October 1994.

JIEO Circular 9008, National Imagery Transmission Format Standard (NITFS) Certification Test & Evaluation Program Plan, 30 June 1993.

APPENDIX I
DOCUMENTATION STATIC REVIEW RESULTS

I-1 General. This section contains comments collected during the initial documentation static review which were reported to the originating agency.

I-2 Documentation Static Review. Tables I-1 through I-3 list comments provided by the JITC.

Table I-1 31 May 1995 JITC Comments

ITEM	PAGE, PARA:	COMMENT	RESOLUTION
Figure B4	30-2, 30.2.3	The upper block should indicate JPEG decompression. Suggest changing the label to read "12 Bit JPEG IDCT DECOMP". Rationale: Clarity.	Comment was incorporated into text.
Table B5	30-4, 30.2.3 .2.2.1	The CEL field units should be Numeric or Alphanumeric. The value in the CEL field should be the number of data bytes that follow the CEL field to the end of the tag. This also applies to all of the other tables in the document. Rationale: Correctness.	Units column was dropped from table.
Table B5	30-4, 30.2.3 .2.2.1	The MAP -SELECT **field** under FORMAT you indicate 0x00, a hexadecimal representation; if this is true, then the RANGE should be 0x00 for method zero. Rationale: Correct range.	This field is now numeric. Its value is '0'.
Table B5	30-4, 30.2.3 .2.2.1	The BAND NUMBER field? Is this also a hexadecimal representation, like the MAP-SELECT **field,** or a alphanumeric representation? Rationale: Correct format representation.	This field is now numeric with a range of 000 to 999.
Table B5	30-4, 30.2.3 .2.2.1	For Units column, the unit BYTES and INTEGER VALUE aren't clear to me. I suggest using Alphanumeric, Numeric and Binary (aka hexadecimal). Rationale: Clarity.	This column was dropped. Example values in the range were more explicit.
Variable Definitions	30-5	The definition for the IY variable is missing. Rationale: IY should be defined.	Comment was incorporated in to text.

Table B-6	30-6	What is the TABLE-ID for? Rationale: TABLE ID should have a purpose.	Comment was deemed inappropriate.
Table B-6	30-6	The COUNTS units, are those in Alphanumeric or Binary? Rationale: Need to specify alphanumeric or binary format.	This column was dropped. Range column is more explicit
Table B-6	30-6, 30.2.3 .2.2.3	Since SI is not used during method 1, why include it in the tag? Rationale: Unused element.	Missing information in original RFC. S1 is now used in Method1.
Variable Definitions	30-8	Does the expression "exp(IY/B)" equate to "$ey^{IY/B}$" or "$10^{IY/Bit}$"? Rationale: Clarity.	Text was modified to show exp() = e^x
Variable Definitions	30-8	The line with the expression IXX = int((IY/ISF) + 0.5 is missing a closing parenthesis. Rationale: Correction.	Text was modified to correct equation.
Variable Definitions	30-8 and 30-13	The value ISF is used in the expression for IX on page 30-13; but OSF is used in the expression for IX on page 30-8. Are these both correct? Rationale: Correct variable term.	Text was modified to show ISF*OSF in both methods.
Variable Definitions	30-8	The expression IX + int((IMIN/OSF) + *0.5)*, I believe, is missing the leading terms "OX = ". Rationale: Missing terms.	Comment was incorporated in to text.
Variable Definitions	30-8	Bottom of the page: The expressions for ISF and OSF; I believe the first S on the right hand side of the equal sign should be changed to a "2". Rationale: Correctness.	Comment was incorporated in to text.
Table B-8	30-10	The UNITS marked as INTEGER- are these binary or numeric? Rationale: Clarity.	Units column was dropped.

Table B-8	30-10	You may want to state that X_1 > 0 and X_1 < X_2 and X_2 < 4096; so there is no confusion as to which segment of the Amplitude Mapping Function to use at the boundary points. Rationale: Clarity.	Comment was not incorporated in to text.
Variable Definitions	30-13	The expression at the top of the page, $X_{j-1} \leq IY < X_j$ seems to be in conflict with the statement in the middle of the page that says "... and Xj is segment (J)'s lower boundary. " Rationale: Clarity.	Conflicting 2nd phrase was dropped from text.
Variable Definitions	30-13	Middle of the page: the equation for IZ, Shouldn't that read " IZ = IY - Xj "? Rationale: Correct algorithm.	Comment was incorporated in to text.
Variable Definitions	30-13	Sentence in the middle of the page that starts " The output of the polynominal mapping function (IX) shall be scaled ". shouldn't IX be changed to IXX. Rationale: Correct variable term.	Comment was incorporated in to text.

Table I-2 19 December 1995 JITC Comments

ITEM	PAGE, PARA:	COMMENT	RESOLUTION
Table B-11	30.2.6	NBLOCA: Notes 1,2 and If: this is a controlled tag. All of its entries should be in clear ASCII text. Rationale: Clear ASCII text is required.	Comment deemed inappropriate. All numeric fields are 4 byte binary integer. Tag is controlled.
Table B-11	30.2.6	NBLOCA: CEL field. Appears to be a conflict between the min CEL value and note 3 (Number of Blocks for which offsets are listed) for this Table. If the min number of blocks can be "1", then CELDATA will only have 8 bytes; 4 for the FRAME 1 OFFSET field and 4 bytes for the NUMBER-OF-FRAMES field. This makes the min value for CEL = to "8". Rationale: Clarity.	Comment was incorporated in to text.
Below Table XVII	5-2, 5.2.3.3. 5.5.3	APP8 Minimum Value: For the following text "an APP8 application data segment should exist for each image block or frame compressed with the 12-bit JPEG algorithm"; Should the verbiage be "exist" or "must exist"? If it doesn't exist, what should the receiver of such a file do? Rationale: Clarity.	The words "must exist" were incorporated into the text.
Table XVII	5-2, 5.2.3.3. 5.5.3	APP8 Minimum Value: Suggest you change the table's title so it includes the number "8"; So it reads " NITF JPEG APP8 Segment Format For............". Rationale: Clarity.	Text was modified to show the accepted name for the APP. APP6/0001.
Table B5	30-4	IOMAPA If this is a controlled tag, all of its entries should be in clear ASCII text. Rationale: Clear ASCII text is required.	Text was modified for Method 0, Table 5 entrys are all clear text. Tag is controlled.

Variable Definitions	30-6, 30.2.3 .2.1.2	IOMAPA, Method 1: "shall be defined as:" section. The IXX line that reads "Scale the original data to 12 bits"; shouldn't that read, "Subtract minimum value from original data then scale difference to 12 bit." Rationale: Clarity.	Text was modified to incorporate comment.
Variable Definitions	30.2.3 .2.1.3 and 30.2.3 .2.1.4	IOMAPA, Input Amplitude Mapping Method 2 and Input Amplitude Mapping Method 3: Where section. The IXX term is not defined as the IX and IXXX terms are. Rationale: Need consistent term definition.	Commment was incorporated in to text.
Variable Definitions	30.2.3 .2.1.4	IOMAPA, Input Amplitude Mapping Method 3: where section. The line that says "IY Re-mapped output pixel value"; shouldn't that say "IY Re-mapped Image Pixel Data"? Rationale: Clarity.	Commment was incorporated in to text.
Variable Definitions	30.2.3 .2.2	IOMAPA: Functionality of NITF JPEG/DCT Expander When Utilizing the IOMAPA Tagged Record Extension, second para says "The second operation adds the minimum value extracted from the JPEG APP8 application segment to each pixel value.,, but Figure B-4 Output Remapping Process, shows that the second step is the down scaling step and the MIN value addition as the third step. Rationale: Clarity.	Text was modified to incorporate this comment.

Variable Definitions	30.2.3 .2.2.4	IOMAPA, Output Amplitude Mapping Method 3. The Amplitude Mapping polynomial shown in the input process isn't an inverse of the polynomial shown in the output section. input IZ = IXXX - X(J-1) IY = int[a0 + a1*IZ + a2*(IZ**2) + a3*(IZ**3) + a4*(IZ**4) + a5*(IZ**5) + 0.5] output IZ = IY- X(J-1) IXX = int[a0 + a1*IZ + a2*(IZ**2) + a3*(IZ**3) + a4*(IZ**4) + a5*(IZ**5) + 0.5] Rationale: Clarity of algorithm.	Text was modified to show different cofficients for input polynomial vs output polynomial.

Table I-3 8 December 1995 JITC Comments

ITEM	PAGE, PARA:	COMMENT	RESOLUTION
Figure B3	30-1, 30.2.3	IOMAP1 should be changed to IOMAPA. Rationale: Typographic error.	Text was modified.
Figure B3 and B4	30-1 & 30-2, 30.2.3	Suggest you show a block for the "scaling" step in the process. Rationale: Clarity.	Text was modified.
Figure B3	30-1, 30.2.3	"JPEG APP Segment" is missing the subscript, should read "JPEG APP8 Segment" Rationale: Clarity.	Figure was modified.
Figure B4	30-2, 30.2.3	IOMAP1 should be changed to IOMAPA. Rationale: Typographic error.	Figure was modified.
All tables		To be consistent with the format used JIEO Circular 9008 and MIL-STD-2500, recommend all tables in this <u>RFC describing tags be changed to show</u> the <u>following columns</u>(left to right): FIELD listing the ACRONYM for the field. DESCRIPTION listing the ACRONYM spelled out. SIZE listing size in Bytes. FORMAT VALUE - an alphanumeric showing acceptable or actual values that can go in this field. Numeric must have leading zeros. TYPE listing R or 0. Rationale: Table format consistency.	Tables were modified to show Field, Description, Length, Value range, and Type.
	30-5,30.2.3.2.2.2.1	The equation for OX = IXX/OSF2 + int (MIN/OSF) + 0.5); the opening parenthesis is missing. Rationale: Correct algorithm punctuation.	Text was modified.
Table B7	30-7, 30.2.3.2.2.3	In the range column of the CEL field you show a value of "18"; for the data list, this should be "00016". Rationale: Correct value.	Comment was incorporated in to text.

Table B-6	30-6,30.2.3.2.2.2.1	The entrys for INPUT MAP VALUE are shown to be 2 bytes long. Since all fields should be clear numeric text, this field should be 4 bytes long. Rationale: Correct length of fields.	Comment was not incorporated. Field were left as 2 byte integers.
Variable Definitions	30-8, 30.2.3.2.2.3	Where the variables OX, IX, IXX, and IY are described; suggest you list these variables in the order they would be generated during the expansion process. Rationale: Logical listing of variables based on generation order.	Text was modified.
Variable Definitions	30-9, 30.2.3.2.2.3	IMAX = 4096 and IXMAX = IMAX -1. What not show IXMAX=4095; same as was done for Method 1, para 30.2.3.2.2.2.1 Rationale: Consistency.	Comment was incorporated.
Table B8	30-10, 30.2.3.2.2.4	Shows that the polynomial coefficient fields are REAL 4 byte floats. All number entrys in the tag should be in clear numeric text. We recommend coefficients be entered in scientific notation (ie. 4.565 x 10^{+12} could appears as 4565E + 12 or 4.565 + 12 or 4565 + 13 or 4.565X 10 + 12). To determine appropriate size for this field, author of the RFC must nominate a maximum number of significant digits for all of these coefficients. For example, a 10 byte field would allow for 3 bytes for the exponent " + 12" and 7 bytes for the significant digit of the coefficient; " 1234567 " (the decimal point could be assumed at a given place in the string.) Note: the range of a 4 byte single precision float is 3.4E-38 to 3.4E+38. Rationale: Clarity.	Comment was not incorporated. Fields were left as 4 byte real.

Table B8	30-10, 30.2.3 .2.2.4	Note #:4 IEEEE should be IEEE. Rationale: Typographical error.	Text was modified.
	30-13, 30.2.3 .2.2.4	End of para, Method 3, Does not contain a "Where" section. At the end of Method 2 and Method 1 section there is a "Where" paragraph that gives a one line description of all variables that are used in that section. Suggest you add that type of section to Method 3. Rationale: Clarity.	Text was modified to incorporated comment.
Para 2, line 2	30-13, 30.2.4	Change "HHECCA" should be "NHECCA". Rationale: Typographical error.	Text was modified.
Para 2	30-14,	Suggest you change the phrase "to the NITF header" to "of the NITF Header". Rationale: Correct phrase.	Comment was not incorporated in to text.
Para 2	30-16, 30.2.5	Change “controlled tagged record extension" to "registered tagged record extension". Are both NIECCA and NHECCA Registered tags? Rationale: Correct term needs to be used - controlled tagged record extension or registered tagged record extension.	Text was modified to show them as Registered
Para 2	30-16, 30.2.5	Suggest you change the pharse "to the image subheader" to "of the image sub Header". Rationale: Correct phrase.	Comment was not incorporated in to text.

	30-17, 30.2.5 .2	This sentence implies that the image subheader and only controlled tags go through the FEC. Suggest you change the wording so that the entire image subheader, with controlled and registered tags (excluding the NIECCA tag) goes through the FEC process. Rationale: Clarity.	Comment was not incorporated in to text.
	30-15 & 30-17, 30.2.4 .2 and 30.2.5 .2.	Need period on last paragraph last sentence. Rationale: Correct punctuation error.	Comment was not incorporated in to text.
	1, 30.2.6	First sentence isn't clear. Do you mean to say that this tag lists the offset of each image block relative to a unique start point in the file or does it list the number of bytes between the start points of the consecutive image blocks? Rationale: Needs clarification.	Comment was not incorporated.
	1, 30.2.6	2nd paragraph. Is NBLOCA a controlled or registered tag. Rationale. Controlled tags should show their data stored as clear readable alphanumeric text.	This tag is controlled with some fields set to binary values.
	5-1, 5.2.3. 3.5.5. 3	NITF APP8 Image Block Min Value, Fig. XVII; change "BorP" to "B or P". Rationale: Typographical error.	Text modified.
	5-1, 5.2.3. 3.5.5. 4	NITF APP9 Image Block Min Value, Fig. XX; change "BorP" to "B or P". Rationale: Typographical error.	Text modified.

	5-5	RFC for APP8 and APP9, figures XVII, XIX, XX and XXI conflict among placement of APP segments. The figures XVIII and XIX show APP8 coming after APP7, while figures XX and XXI show APP7 coming after APP8. Do you mean to say that the order of these APPs is dependent upon the exsistance of APP9. This may be a question for a higher authority, "should pacement of all APPS, except APP6, be a free for all or should they have a fixed order." Rationale: Clarity.	
	5-5, 5.2.3.3.5.5.4	NITF APP9 Data Segment. Suggest that you explicitly state that "for purposes to FEC-1 encoding, the JPEG header includes the application data segments (excluding the APP9 segment)", or does it? At the Frame or Scan level exactly what part of the Frame or Scan JPEG header is FEC encoded? Rationale: Clarity.	Text was modified to incorporate comment. FEC encoding excludes the APP9 section.
	5-5, 5.2.3.3.5.5.4	NITF APP9 Data Segment. APP9 comes in a type 0 variety signalling that no ECCs were developed. Does this mean that an APP9 should always be included in a JPEG file; when no ECCs are calculated. Or has the author defined a APP9 variety that will never be used? Rationale: Clarity.	Text was not modified. Type 0 is allowable.